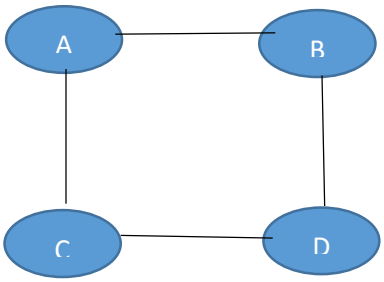
RUNNING THE CODE

The following command line arguments are used:

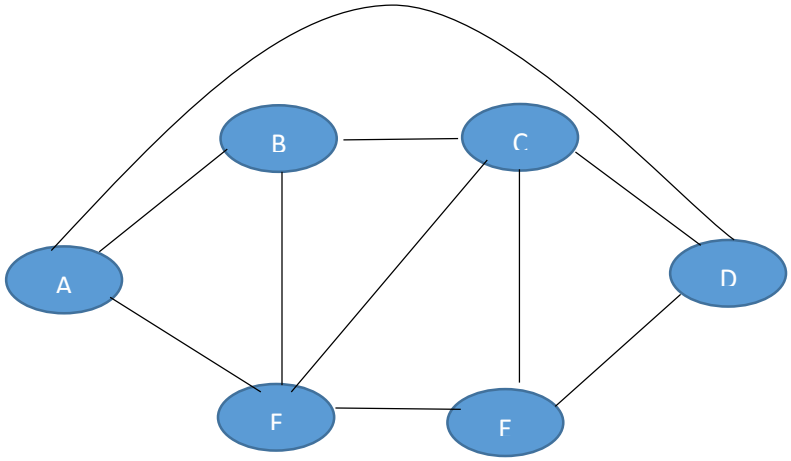
1. Configuration file
2. Port (65529)
3. TTL (Default: 90)
4. Infinity
5. Period (Default: 30)
6. Split-horizon (0 or 1)

GRAPH

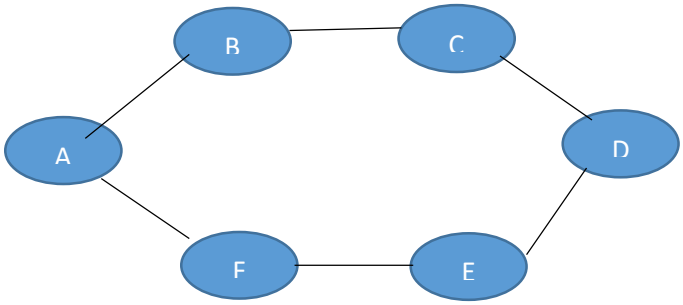
Below are the graphs for which we tested our distance vector program convergence and node failures.



Graph 1



Graph 2



Graph 3

COST MATRIX AND ROUTING TABLE

Routing table is computed based on a cost matrix. Cost matrix is computed based on the configuration file provided in the command line argument. Below is a sample configuration file for Graph 1 Node A:

129.79.242.134 (B) yes

129.79.242.136 (C) yes

129.79.242.138 (D) no

The initial routing table from the cost matrix is computed as an array of below structure:

```
struct RouteEntry{
        char destination[16];
        char nextHop[16];
        int cost;
        int ttl;
}
```

MULTI-THREADED APPLICATION

We have implemented multi-threaded application where a separate thread is created for receiving, updating the routing table and sending out triggered updates. The main thread creates the initial routing table and sends out periodic advertisements to all neighbors.

POSIX thread mutex lock is used to ensure consistent use of the shared variables among the above two threads. The shared variables are: the routing table structure array, number of nodes, infinity, neighbor port, split horizon. The conditional variable used is pthread_mutex_t lck which prevents other threads from executing until current thread completes its execution.

CONVERGENCE AND NODE FAILURE:

Tested on graph 3 for period as 30 seconds and default TTL as 90 seconds and infinity as 999.

Without Split Horizon:

Normal Convergence	Convergence after single node failure	Convergence after double node failure
45 seconds	167 seconds	197 seconds

With Split Horizon:

Normal Convergence	Convergence after single node failure	Convergence after double node failure
49 seconds	181 seconds	212 seconds

With split horizon, it is observed that the time taken to converge is more.

INFINITY AND CONVERGENCE TIME

We tested the code for different values such as 16,50 and 999 and it was observed that there was no change in the convergence times as we did not encounter count to infinity problem.